CLAIMS

What is claimed is:

1   1. A method for providing access to a resource in a programming environment

2   supporting concurrent tasks, comprising the steps of:

3   (a) providing a latch to a first task, wherein the first task requests the latch to obtain

4   access to the resource;

5   (b) releasing the latch after the first task has accessed the resource, while still

6   holding the latch;

7   (c) marking the latch stealable, while still holding the latch; and

8   (d) marking the latch unstealable if the latch is not marked stolen by a second task

9   seeking access to the resource.

1   2. The method of claim 1, wherein the providing step (a) further comprising the

2   steps of:

3   (a1) providing to the first task a latch marked stealable and held by a holding

4   task; and

5   (a2) placing the resource in a consistent state prior to the first task accessing

6   the resource.

1   3. The method of claim 2, further comprising the step of marking a set of flags for a

2   latch stealable, stolen or unstealable, as requested by the latch.

1   4. The method of claim 1, further comprising the steps of:

2   (e) assigning a priority to a stealable latch; and

3   (f) requesting a latch with a selectively determined priority, whereby a stealable

4   latch is provided to a requesting task having a determined priority above a threshold value

5   defined by the assigned priority for the stealable latch.

1   5. A computer readable medium containing program instructions for providing

2   access to a resource in a programming environment supporting concurrent tasks, the

3   program instructions for:

4   (a) providing a latch to a first task, wherein the first task requests the latch to obtain

5   access to the resource;

6   (b) releasing the latch after the first task has accessed the resource, while still

7   holding the latch;

8   (c) marking the latch stealable, while still holding the latch; and

9   (d) marking the latch unstealable if the latch is not marked stolen by a second task

10   seeking access to the resource.

1   6. The computer readable medium of claim 5, wherein the instructions for providing

2   further comprising instructions for:

3   (a1) providing to the first task a latch marked stealable and held by a holding

4   task; and

5   (a2) placing the resource in a consistent state prior to the first task accessing

6   the resource.

1   7. The computer readable medium of claim 6, further instructions for marking a set of

2   flags for a latch stealable, stolen or unstealable, as requested by the latch.

1   8. The computer readable medium of claim 5, further comprising instructions for:

2   (e) assigning a priority to a stealable latch; and

3   (f) requesting a latch with a selectively determined priority, whereby a stealable

4   latch is provided to a requesting task having a determined priority above a threshold value

5   defined by the assigned priority for the stealable latch.

1   9. A latch mechanism for a programming environment supporting concurrent tasks,

2   comprising:

3   means for providing a latch to a first task, whereby the first task holds the latch;

4   means for the first task to release the latch while holding the latch;

5   means for the first task to mark the latch stealable; and

6   means for the first task to remark the latch unstealable where the latch is not marked

7   stolen by a second task seeking access to the latch.

1   10. The latch mechanism of claim 9, wherein the means for providing a latch to the

2   first task further comprising:

3   means for the first task to be provided, on request, with a latch marked

4   stealable and held by a holding task, and means for placing any resources associated

5   with the latch in a consistent state prior to the first task accessing the resources.

1   11. The latch mechanism of claim 10, further comprising a set of flags for marking a

2   latch stealable, stolen or unstealable, by a task.

1   12. The latch mechanism of claim 9, further comprising:

2   means for assigning a priority to a latch marked stealable; and

3   means for a task to request a latch with a selectively determined priority, whereby a

4   stealable latch is provided to a requesting task having a determined priority above a

5   threshold value defined by the assigned priority for the stealable latch.

1   13. A computer readable medium containing program instructions for implementing

2   an exclusive latch mechanism for associated resources in a computer programming

3   environment supporting concurrent tasks, the program instructions for:

4   providing a latch to a requesting task, whereby the task becomes a holding task and

5   the latch becomes a held latch,

6   releasing the held latch, whereby the task ceases to be the holding task,

7   marking as stealable the latch held by the holding task, the means to mark the held

8   latch stealable comprising means to specify a flag and a cleanup subroutine, each being

9   associated with the held latch and the holding task, and comprising means to set the flag to

10   the value stealable,

11   marking the held latch unstealable, the program instructions comprising:

12   determining the value of the flag associated with the latch, and

13   setting the associated flag to the value unstealable, where the determined

14   value of the flag is stealable,

15   the program instructions for providing a latch to a requesting task, wherein the latch

16   is a held latch, further comprising instructions for:

17   determining the value of the flag associated with the held latch and the

18   holding task, and

19   providing the held latch to the requesting task where the flag value of the

20   held latch is stealable, the providing instructions further comprising:

21   executing a cleanup subroutine associated with the held latch and the

22   holding task, and

23   setting the flag associated with the held latch and the holding task to

24   the value stolen.

1   14. A computer readable medium containing program instructions for implementing

2   a full function latch mechanism for associated resources, the program instructions for:

3   providing a latch to a requesting task, whereby the task becomes a holding task and

4   the latch becomes a held latch,

5   releasing the held latch, whereby the task ceases to be the holding task,

6   maintaining a plurality of flags, each flag in the plurality of flags being associated

7   with the held latch and the holding task,

8   marking the latch held by the holding task stealable, the marking instructions further

9   comprising:

10   specifying a cleanup subroutine associated with the held latch and the

11   holding task, and

12   assigning the value stealable to the flag associated with the holding task and

13   the held latch, and

14   marking the latch held by the holding task unstealable, wherein the marking

15 instructions further comprising:

16   determining the value of the flag associated with the holding task,

17   indicating that the determined flag value is stealable, and

18   setting the associated flag to the value unstealable, and

19   determining the value of the flag associated with the holding task and for indicating

20 that the value of the associated flag is stolen where the determined value of the flag is

21 stolen,

22   wherein the providing instruction further comprising instructions for providing a

23 held latch to a requesting task, comprising:

24   determining the value of the flag associated with the held latch is stealable,

25   executing a cleanup subroutine associated with the held latch and the holding

26   task,

27   setting the flag value to stolen, and

28   providing the held latch to the requesting task.

1   15. The computer readable medium of claim 14, further comprising instructions for:

2   maintaining a count of the number of tasks holding a specified latch,

3   maintaining a count, for the specified latch, of the number of tasks that have marked

4 the latch stealable, and

5   comparing the count of the number of tasks that have marked the latch stealable with

6 the number of tasks holding the specified latch.